CGRA354 Computer Graphics Programming T1/2024

Assignment #3: Animation/Boids (15 points)
Assigned: Tuesday, 22nd April
Due: 16th May 2024 11:59pm

The areas addressed in this assignment include the handling of vector algebra in order to compute forces, average positions, velocities, distances and proximity determination and its use in Computer Graphics for Object Animation.

Codebase

You are provided with a cross-platform programming framework specifically designed for the Computer Graphics assignments. The framework performs the initialization of the OpenGL pipeline allowing to focus on the assignment tasks. Comprehensive instructions for building the framework on different platforms can be found in the `README.md` file located in the source code root folder.

Turn in procedure

You should submit your work as a zip file using the ECS submission system. Please name your file as `<LastName><FirstName>-Assignment<X>.zip` where X is the assignment number. When your file is unzipped you should have:

1. The C++ code and programs you have written. You should use files in the form of the samples given, rather than producing files from scratch. This will help us follow your code.

2. Sample images created by your program. You can save these by taking a screenshot or use a screen capture tool.

3. A readme.md, txt, pdf or MS Word file that answers any questions posed, and that lists the input used to create the images you include. Alternatively you can provide a narrated screen-capture video showing the functionality of your program.

4. Any other information or supporting documentation to help us run and evaluate your submission.

If your programs fail on the machines used for grading, you may be asked to bring in your system to demonstrate that the files you submitted functioned in the environment you worked in.

Grading rubrics

There are multiple questions in this assignment, with some question having multiple subparts corresponding to a particular tier (core, completion, challenge and writing).

Part 1: Animation/Boids (15 points)

In this exercise, you will implement the boids flocking algorithm, originally developed by Craig Reynolds. Each boid is controlled independently but the behaviours programmed into the algorithm mean that group behaviour emerges from the individual behaviours. The later stages of the assignment build extra features on top of Reynold's basic algorithm.

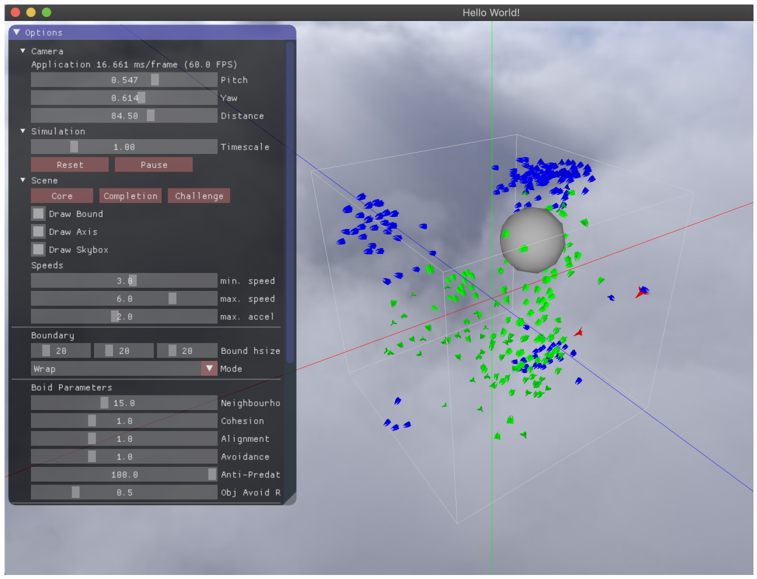

Figure 1: Boids simulation running within CGRA251 Framework.

Core (8 points)

Get a single flock to work. You will create a single flock of boids. Each boid will exhibit the five standard boid behaviours:

1. **Avoidance:** steer to avoid crowding local flockmates.
2. **Alignment:** steer towards the average heading of local flockmates.
3. **Cohesion:** steer to move towards the average position of local flockmates.
4. **Confinement:** keep the boids within a particular area, so that boids stay on the screen rather than fly into the distance.
5. **Sensible speed:** keep its speed between a minimum and a maximum, to emulate a real birds that cannot fly slower or faster than certain speeds.

Provide ImGUI sliders that control the following factors: the strength of the avoidance, alignment, and cohesion behaviour, the size of "local", which is the distance that a boid can sense the positions and headings of its local flockmates.

Note: You should start by getting a few boids to work with behaviours 4 and 5, then extend this to a flock of 100–300 boids exhibiting all five behaviours.

Completion (5 points)

Two flocks and a predator:

- You will create two flocks of boids, distinguished by different colours. A boid will **avoid** all other local boids but it will **align** and **cohere** only with local boids in its own flock.
- Add a predator to the simulation. The predator frightens the boids and also tries to capture a boid.

Note: You will need to change the boids' behaviour so that there is an extra force that drives them away from the predator. You need to think about how far away boids should be allowed to perceive a predator. The predator is going to have some similar and some different behaviours to the boids. Instead of the alignment and cohesion behaviours of a boid, the predator acts alone and hunts down boids. You need to determine how the predator behaves to catch its prey. For example, how far away can a predator see prey? Does the predator fixate on a single boid and chase it to the exclusion of all other boids? Does the predator chase the nearest boid and switch its attention if another boid gets closer? Is there some more complex algorithm that allows a predator to be more successful?

Challenge (2 point)

Things to avoid:

- You will create other objects for the boids to avoid that are significantly larger than a single boid so the simple boid avoidance method will not work.
- You should demonstrate at least three large objects. For example, you may place three or more large spheres in the scene.

Note: You must use the OpenGL Mathematics (GLM) facilities on this assignment.

References

[1] Wiki: Boids,
`https://en.wikipedia.org/wiki/Boids`

[2] Stanford University: Modeling Natural Systems,
`https://cs.stanford.edu/people/eroberts/courses/soco/projects/2008-09/modeling-natural-systems/boids.html`

[3] Craig Reynolds: Boids (Background and Update),
`https://www.red3d.com/cwr/boids/`